



TITLE:

物(もの)性研究(<特集>複雑系の展望-複雑系若手の立場から)

AUTHOR(S):

山本, 知幸

CITATION:

山本, 知幸. 物(もの)性研究(<特集>複雑系の展望-複雑系若手の立場から). 物性研究 1997, 68(1): 116-123

ISSUE DATE:

1997-04-20

URL:

<http://hdl.handle.net/2433/96011>

RIGHT:

エッセイ
物(もの)性研究

山本知幸

東京大学大学院総合文化研究科広域科学専攻 相関基礎科学系¹ E-mail:

yamamoto@complex.c.u-tokyo.ac.jp

1 はじめに

ここでいう「物」とは、自然界に存在する物もそうであるけれども、計算機で実行されるプログラムの様なある媒体の上のみ存在する物も含んでいる。つまりは、頭の「外」にある物であり、それゆえ名前を持っていない——名前は頭の中にあるから。もちろん、我々はコンシステントな辞書を持っているので不便を感じない。しかし、新しいものを創造するときにはそのシステムの綻びが露わになる。例えばその綻びを楽しんでみようと思う。

自分のことを少し振り返って見ると、思いもよらないようなことを考えてみたくて、複雑系に引き寄せられて来たように思う。昔からSFを読み漁ったりしているうちに、単に想像もつかないことと言うだけでは面白くなって、科学の厳しいルールはあるがその強力さゆえに逆手にとれば武器になるという点に魅かれてきた。それを認識したのはホフスタッターの「ゲーデル・エッシャー・バッハ」など[1][2]だったが、そこからカオスを経て複雑系まで行きつくのにさして時間はかからなかった。

また、どこか根本的な所で人間の思考というものを信用していないところがある。自分の使っている言語や概念などの道具によって思考が誘導されたりすることはあるが、むしろそれらとの相互作用の間にこそ本質があるような気がしてならない。計算機でモデルをつくって、走っているプログラムをあれこれいじりながら研究を進めるという態度にも魅かれたのだと思う。

自分の興味は、研究分野でいうと人工生命に位置している。生きているという状態を知りたいからということがその理由であるけれども、その裏にはいまある人工物の限界を超えたものを作り出したいという欲求がある。人工物の限界を超えたところを想像しようとするのであれば、おのずと手本は自然物になるから。

手本はそのままコピーされるべき物ではなく、どこかの時点でそれと離れて別物が出来上がるわけで、その点に創造が宿る。しかし手本をどう設定するかによって、到達できる領域が暗に決定されてしまう。それゆえに、つき合い方を考えておくことは研究の要であると思う。

すでに我々は言葉や記号、表、論理などの道具を持っているわけで、これらの道具とのつき合い方も考えねばならない。手本と同様に、自分の思考の範囲を決定するものである。「自由に考える」とは、それらの枠内に留まることに過ぎない様にも思える。近代知性が、いかに「表的思考 = 世界を

¹197年4月より北海道大学理学部

記号にまで分解し、卓のうえに並べて一望の下におき、浮かび上がってくる関係をもとめること」に縛られていて、そしてそれが根深く強力であるかということを知らされたのは、高山宏の「終末のオルガノン」[3][4]をはじめとする評論集だった。この特集の編集者が以前に企画した、94年12月に統計数理研究所で開催された研究会「動的システムの構築と記述に向けて」の講演者の一人でもあるが、参加者はアクの強い喋りも加わって強烈な印象をもったはずである。

ルネサンス以降の暴走した知性の実如何に「表」という道具に因われていたかということを経験として、どこかに突破口を求めたいが、いまの所はその準備段階でもたついている。なぜ表的ではいけないか。端的にいうと、それは「表からもれたものは無視される」[3]からである。多様であり、変化し、ダイナミックなものは、到底表には分類しきれない。

2 物的思考へ

生物のことをもう少し考える。細胞の内部がまるで機械のようになっているとは良くいわれるが、これほど機械らしくない「機械」もないだろう。それは確かに部品がまとまってできたものだけでも、それにラベルはついていないし厳密に決まった動作だけをしているものにも見えない。生物と人の作った機械の差は、数多く集めた時に現れる。部品点数の多い機械は、その数やモジュールの数が増えるにしたがって常に事故の危険性が増す傾向にある。

生物の細胞のように、多数の器官(モジュール)が一つのところに押し込められていると、器官同士が干渉しあって崩壊してしまいそうに感じるが、それは容易には起こらない。人がこのような機械をデザインするのはほとんど無理だろう。しかし、器官が自分の都合で勝手に動いてしまうように、本来物理現象は並列的で、複数の部品からなるシステムが望んだ通りの順番で動いてくれるとは限らない。計算機の内部ではとくにそうであるが、順序をコントロールするために、時間・空間・エネルギーとあらゆる面で多くの資源が費されている。生物がそこまでの「無駄」を順序などの調停に割いているようには思えない。もちろん細胞周期などのサイクルがあるが、その順序性はあくまでも緩い形でしかコントロールされているようにしか見えない。

とくに、運動という時間的要因まで考えると問題はより深刻になる。機械が複雑になると、ある「機能」が完了するのに経過する時間を問題にしなくてはならなくなる。途中で他の「機能」と干渉してしまうことなどが頻繁に起こり得る。それでも上手く動いてしまうことが生物の不思議なところである。何故か、機能と構造が「一致」した美しさがある。そのどちらも明確には出来ないのに。人工の機械でも、一部が壊れて暴走する機械はある意味で美しく見えるが、その原因は設計に表れていない「自然な」動作をするからかもしれない。もちろん、そうでない方が多いけれども。

生物の特徴として階層性や自己複製が挙げられるが、階層性のためには階層としてスケールが分離できる程多くの要素が集まっても満足に動作することが出来ねばならず、また自己複製する為には自分自身の記述を読み出すことも重要だが、自己の内部の同期をとらなくてはならない。この数多く集まることが出来る「スケーラビリティ」と機能を実現する際に要求される「階層間の素早いコミュニケーション」は、今の機械という枠組からはひねり出すことは出来ないだろう。そこからじっくり

覗けば、機能は見えるがあくまでもその境界は緩やかなものでしかない。その柔軟さゆえに生物は強靱だと思っけれども。

このように抽象化しようもないものがあるので、

- なにか構造があるが、分節化することが出来ないもの、または
- 状況に応じて意味 / 機能が変わってしまうもの
- リアルタイムに自ら変化するもの

というもの — 定義不能であるがゆえ「物」としか呼びようがないが、モデルの実装ではオブジェクトとして実体があるもの — をもとにした考えを作らなくてはならないと考えた。

これらは論理的記述が出来ないために、よく迂回したり無視したりされるが、ここに本質的 중요さを認めている。物理的な実体でも仮想的なオブジェクトでも同様にこの性質をもつものは構成可能だろう。ただし、こういう「物」をモデルに含んでいると言うことは、必要条件に過ぎない。

連続状態を持つ力学系によって記述される物理的実体 (のモデル) と計算機のような有限状態機械のリアルタイム性を持った結合というイメージが最も近い。というよりも、記述する手段というものをその特徴から判断して分類してゆくと、力学と計算機の2通りになるということである。これらのメカニク的な記述手段をモジュールとして、そこから全体をまとめあげる技術がある筈で、やはりそれはこの2つとは別物であろう。

それはともかく、計算機と力学の差とは状態が離散 / 連続であることと、順序的記述の可 / 否にある。どちらでも相手を真似しようと思えば出来なくはないが、マップを計算機で計算する場合以外はどこかに無理があって、それ以上の発展が出来なくなってしまう。これは経験的な評価ではあるが、媒体の特質と親和する表現でなくては高次のモデルへと進展できないのである。では、ここで力学系と計算機の特徴をまとめることにする。

2.1 力学系

力学系によって初めて明確な形でカオスが意味づけられたように、複雑な運動を表現することができる記述方法である。なお次項と区別するために、ここでは連続力学系のみ考える。短い簡単な記述からでも複雑な挙動を作り出すことができ、また挙動からもとの力学を再構成することは、一般には困難なので先の定義の条件を満たす。しかし、方程式として記述されているものが、直接理解できないのは逆説的である。

力学系は当然のことながら時間に沿った運動や変化を表現する、ダイナミックな表現であるし、幾つかのサイトの間の結合や干渉といったものも楽に記述できる。しかし、一方では順序的なプロセスを作ることが出来ないという問題がある。例えば、干渉などによって現在の状態が崩すことはあっても、それを元に戻すことが出来ない。実際の計算機システムでは順序を飛ばして優先的に行うプロセスを扱う機能が用意されていて、それは割り込み処理と言う形で実装されているが、力学系では無

理だろう。システムに目的を持たせようとするならば、重要な場面を壊さないためにどこかで系の多様性を消さなくてはならなくなる。しかしこれでは力学系の複雑な挙動を生かすことが出来ないことになるので、意味が無い。

ということは、力学系の複雑さを生かしつつも思った通りの挙動をする様なシステムを設計することが極端に難しいということである。何かのモデルを作るとしても「それらしい挙動」を得る以上の「らしさ」—たとえば実物に近付けるために必要なモデルの目的論的変更—も困難だろう。人間の思考とはかなり異質なものを作れる手段ではあるのだが。

2.2 計算機

計算機実験により、我々は自分達の頭に収まり切らないような思考実験が出来るようになった。複雑系における計算機の意義は言うまでもない。有限状態機械として考えると力学系のサブセットに見えるが、状態が有限であるが故にプログラムが可能であると言う点が特徴的である。任意のビットが何の役割をするか、明示的に示す事が出来る。つまり、人間の思考の論理的な部分との親和性が高い。

状態が離散であることによって、外部(プログラムする人間)からの恣意的な意味づけが出来ることはシステムを目的に沿って構成する際の大きな利点である。この手続き的な記述は連続系でも出来てはならないだろうが、ノイズや自分自身の運動によって容易に壊れてしまうだろう。順序的な動作など期待すべくもない。

さて、計算機のプログラムのような手続き的・順序的な記述は人間にとって楽なので、それをつき進めていって銀行のオンラインシステムなどの複雑で巨大なシステムまでもが構築されている。しかし、ここでいう複雑さと、我々が複雑系で目指したい複雑さは異なっている。計算機に限らずとも人間が作った「複雑な」システムは、多くの単純なモジュールからなるが、それらの干渉を出来るだけ防ぐという方針で組み上げられている。それぞれのモジュールが設計された目的通りに動くように。そしてモジュール間の干渉は常に「悪」とされる。

しかし、全ての物事を手続き的に書きあらわそうとするとうまく行かないように、計算機でうまく表現が出来ないものもかなりある。複数のプロセスが並列に実行されるマルチタスクのシステムではデバイスなどの資源の競合が起こり、ひどい時はデッドロックしてシステム全体を停止させる[5]。とくにネットワーク環境では、複数の計算機(プロセス実行の主体)が並列かつ非同期にアクセスしあうので、効率の良いプログラムを書くのは困難になる。

一つのプロセスの立場に立って考えると、相手の状態を知りえないまま呼出をかけなくてはならないので、応答の保証もないまま待ち続けなくてはならない。待っている間にも自分の占有しているデバイスやファイルなどの資源を他のプロセスがアクセスしようとして競合がおきるかも知れない。自分が待っている相手が、自分の抑えている資源によってブロックしていたら即デッドロックになる。それを避けるために一定時間以上待ったらタイムアウトで失敗させるなどの例外処理が絡み合ってしまう。例外処理の途中で例外処理が起きたり、他の例外処理と矛盾していたりする場合は悲惨である。

自分の内部の処理は単純な形でプログラムが書けたとしても、外部や他者が絡んでくると途端に難しくなる。一つのプロセスの中では単一の意味論と時間に統一して矛盾を無くすことは出来る。また、結果の正しさを保証するためにプロセスは独立でなくてはならないが、それ故に相手の出方が読めず、コミュニケーションに関して融通が利かないものになっている。複数のプロセスで共通な状態と時間と意味論を持たせることがどれほど大変か、そしてそれが一度崩れてしまうと何が起きるか分からないということはネットワーク環境のシステム管理を担当したことがある人なら誰でも感じていることと思う。

独立性を堅持しようとする余り、競合によってシステムを崩壊させてしまうのであれば、むしろ干渉で互いの状態が変わってしまうことを積極的に利用した方がいいと思える。プログラムが「物」に変わることを企てているわけで、これはシステム論の再構築ということになるだろうが、何とかして実現させたい。ここまでの議論では、順序性が維持されず、それぞれのモジュールが勝手にプロセスを進めて干渉しあうがうまく動くシステムなんて無理に思えてしまうが、生物とはそういうものである。生物と人工物では設計原理が異なっていることを認めて、その上で満足な比較が成立するレベルまで考えを進めたい。

2.3 結合の技術

上の2つはモデルの要素であった。モデルもメディアであるとするならば、その上で何が表現されうるかを論じておかねば不十分だろう — 文字と言葉は別物とみるように。また忘れてはならないのは、さきにも挙げたがモデルを構成する際に要素をまとめあげる「結合」というデザインの技術である。

あるシステムを作り上げるのに必要な、仕様書をまとめるなど枠組を決めるための作業は、そこでつかわれる個々の技術が分かっているだけでは駄目で、例えば名前付けのセンスや取捨選択のバランス感覚などの別物の技術や知識が必要である。複雑なモデルをつくるのに必要なのは、むしろこちらの側の技術であると考えている。

システムとしてのモデルには、均質な要素の組合せで出来るものでもない限り、複数の種類の要素がモジュールとして詰め込まれる。その中には、モジュールが独立性を持っていることの結果として、設計者の各々の要素に対する「意図」としての意味論(セマンティクス)が複数存在する。また、モデルとして組み合わせられる以上、内部でモジュール間に相互作用があるのは必然である。それによって実際は元と異なった意味論(のようなもの)が出来ているはずである。しかし、それをそのまま読み出すことも、厳密な予測をすることも難しい。

モデルにおいては予測し得ない挙動を見出すのが楽しみなのであるが、それが「機械」として実装されるものであればそうは行かない。動作が把握できないもの、安全性が無いものは危険であるし実用にもならない。しかし、把握できるシステムとは、要素の性質の単純な線形結合的な和で表現できるものでしかないので、カオスのような予測困難な状態や自発的な発展は許されない。とはいえ、

把握できるような性質の良いものは頭の中にしか存在しない²。確かに、機械というものは設計された通りの動作をするように作られている。しかし、その設計が正しいかどうかは、そのままでは分からない。要素の自然な挙動を、外から加えた目的に強制的に従わせるような方針のデザインでは、これ以上複雑なものを作れないように思える。

しかし個々の要素技術は発展し続けているから、把握の出来ない程複雑なものがいくらかでも生み出されている。これは、危険な状況に他ならない。機械のどこかが壊れて、その構成要素の物理的な意味での本来の挙動が現れた時にはなす術を知らないだろうから。機械と長く付き合っていれば、経験的な知識によって—それがどう獲得されるかは分からないのであるが—把握は出来るようであるが、これは個人的なものであって、他人に伝えることは難しい。とくに対象がダイナミックであれば論理的な記述が出来ないから。

動的なものの記述は、力学系のところでも触れたように、今のところ確立された方法はない。うまく記述できなければ、対象を外部的な力で強制的に望む状態に持って行くことは不可能になるので、インターフェイスや制御という概念を変革しなくてはならなくなるだろう。それはデザインという技術の全体的な見直しにつながる。

3 開いた体系—メディアとしてのモデル

自己完結的な体系こそ美しいと、我々は考えて来たようである。とくに自然科学はそうである。先にふれた「表」との関連でいうと、表に展開された体系を意味あるものにしようと思えば「矛盾のないこと」「表から洩れたものをなくすこと」になるので、必然的に完結することが要求される。しかし、全ての事物を網羅した表(つまりは完全言語)の構築に成功した例はない[6]。

一般には記述体系が完結していないならば、開放的な記述体系というものは必要無いと思われるかも知れない。しかし、前提として開放性を認めていない体系では、矛盾が起きた時に容易に崩壊してしまう。「開いている」とは、自ら発展できるシステムのことである。時と共に状況に応じて、それを構成する「もの」の意味が変わってしまう。また、そのとり得る可能な状態空間全体は膨大で調べ尽くすことは不可能であるから静的な表現はできない。この様なシステムは、別に特殊な場合でなくとも構成可能である。開放性とは、その表現力が貧弱であるためではなく、その強力さ故に獲得されるものである。むしろ閉じてしまうことは、部分空間に引き込まれてしまって出て行かなければ良いので、常に可能かつ容易である。

この「開放性」には、「物」という側面が重要な役割を担っている。例えば、完結した記述システムの一つである、論理という記述方法では、記号を変えたり増やしたりせずに意味の変化しうる要素をつくることができず、構造を変えずに意味だけがかわってしまうという発展性は見出されない³。論理的なシステムには、一寸した摂動で異なる挙動が導かれるという、「意味のひねり」は含まれない

²実世界では、どんな機械でも使い方によっては危険を招くが、そのことはPL法の施行以降にあらゆる製品に警告表示が貼られたことにより端的に示される。

³論理を含んでいる自然言語では比喩や換喩などで多重的な意味体系が実現されているが、この差異は興味深い

い。そのような厳格な体系でも、それとは別の意味論で解釈できればその「ひねり」を実現出来るかも知れない。論理の表もまた外部からは「物」として見られるべきであるという意味で、(強引に一般化すると)意味論や価値観の多様性は非常に重要であると考えることができる。

相互作用が絡み合うなかで、局所的に構造が作られて固定化される。それらの構造間の相互作用というレベルがつくられれば、そこからまたさらなる構造の発展を期待できる。ただし、力学系の様な実数変数を用いた記述方法では固定化するということが難しく、論理的記述と適度な組み合わせで適度な解像度を作れるようにすると言うのが、今の所の最良な方針だと思われる。また、一つの文脈に固定化されことを防ぐために、プロセスの主体を複数用意して対立させる必要がある。個々のプロセスは孤立しているときは非力であるとしても、相互作用を上手くデザインしてリアルタイムのフィードバックを用意して、そのレベルでの複雑さを獲得することにより発展することはあり得ると考えている。

複雑なシステムは自然界には一般的に見られる — というよりも、自然には多くの要素が場所を同じくしていて、何が「壊れている」状態であるか分からないといった方がいいだろう。少なくとも、崩壊を起こしたり止まってしまうことはないように思える。しかし、それぞれシステムとしての、何らかの構造がある。その構造ないし機能が無意味であるのならば、残ることなく消えてしまうだろう。一見無駄なものが多いように見える自然界であるが、ゴミの少なさを見れば、その効率性がわかるだろう。むしろ、シンプルな機能を作ろうとする人間の試み自体が多くのゴミや無駄を生産している。エントロピー的な見方での、総合的なコストは無視できない。少ないゴミでシステムを運用する事は、我々の想像を超えて難しい事であるが、無駄が多いような生物が達成しているという事はどのような事なのか、考え直さねばならない。しかし、最も重要な部分は相互作用のネットワークの海の中に隠れていて見えないようになっている。

4 おわりに

人間の思考というものも同様に、個人が完全であって無限の能力を持っているというのはいない。古くから文字や、図表などを使って脳の力を拡大してきた。それは他人に伝えるためのみならず、自分とのフィードバックのためでもある。言葉と文字は別物であるから、メディアの上に固定化したところに、意味のずれが生じる。それを読みとって、ひねりを加えて仕立て上げるという創造が生まれることがよくある。作品が完成する直前の、意味が何度もダイナミックに反転するスリルは何事にも代えがたい。

これは、不完全な記述とのリアルタイムのフィードバックによって実現されるものである。他の人とのディスカッションの効用も同様である。勿論、意味の無い苦痛に満ちた会議の方が容易であるのも事実であって、それはモデルを作る際にも当てはまる。計算機と言うメディアによって、これまでは形にならなかったプロセスも試作ができるようになった現在、よりメディアとの付き合い方が問題になってくる。創造性を加速するインターフェースというものはなかなか見当たらないから。

例えば、メディアとしての計算機を最も活かせる筈の人工生命でも、一時の熱狂を過ぎると同じ

ことの繰り返しになっているのがほとんどである。ほんの数例の面白いものを除いては、「画が綺麗になった」という評価しか聞かない。しかし、創造へと導くために綺麗であるのであれば歓迎されるべきである。

最後にこの文を書いた過程を振り返ってみると、残った部分はほとんど手書きのメモをもとに入力されたもので、最初からタイプした部分の多くは、削られてしまっている。論文やプログラムはほぼ全て、簡単なスケッチをもとにエディタで打ち込むので、タイプしながら考える事ができない訳ではない。まだ計算機と言うメディアとの付き合いが上手くないからなのかも知れないが。

手で書く利点は消した部分も斜線の下に残っている事で、プロセスとして自分がどう考えているのか把握できる。勿論、それだけが強力というのではなく、タイプセットされた出力と対比することが必要なのである。同じ所を交互に何度も行きつ戻りつして読んでゆくと、異なるメディアのつきあわせによって、主題と変奏の反復のなかで、直すべき所が浮かび上がって来るようである。

メディアの違いによる微妙な意味のゆらぎを捉えて、そこから情報を引き出しているということと思うが、外からみると儀式にも思える所為である。複数のプロセスを処理する主体がそれぞれの意味論で互いに干渉しあうところから出発するわけだが、今のところそこまでしか我々には行きつけないでいる。

参考文献

- [1] D. R. Hofstadter. *Gödel, Escher, Bach: or an eternal golden braid*. Basic Books, 1979. 邦訳「ゲーデル、エッシャー、バッハーあるいは不思議の環」野崎和弘, はやし・はじめ, 柳瀬尚紀 訳 1985 白揚社.
- [2] D. R. Hofstadter. *Metamagical Themas*. Basic Books, 1985. 邦訳「メタマジックゲーム」竹内郁雄他訳 白揚社 1990.
- [3] 高山宏. 終末のオルガノン. 作品社, 1994.
- [4] 高山宏. 魔の王が見る. ありな書房, 1994.
- [5] Andrew S. Tanenbaum. *Modern Operating Systems*. Prentice-Hall, 1992. 邦訳「OSの基礎と応用」1995 トッパン.
- [6] ウンベルト エーコ. 普遍言語の探求. 平凡社, 1994.